

## **Development of Network-based Communications Architectures for Future NASA Missions**

**Richard Slywczak**  
**NASA/Glenn Research Center (GRC)**  
**Cleveland, Ohio**  
**Richard.A.Slywczak@nasa.gov**

### **ABSTRACT**

Since the Vision for Space Exploration (VSE) announcement, NASA has been developing a communications infrastructure that combines existing terrestrial techniques with newer concepts and capabilities. The overall goal is to develop a flexible, modular, and extensible architecture that leverages and enhances terrestrial networking technologies that can either be directly applied or modified for the space regime. In addition, where existing technologies leaves gaps, new technologies must be developed. An example includes dynamic routing that accounts for constrained power and bandwidth environments. Using these enhanced technologies, NASA can develop nodes that provide characteristics, such as routing, store and forward, and access-on-demand capabilities. But with the development of the new infrastructure, challenges and obstacles will arise.

The current communications infrastructure has been developed on a mission-by-mission basis rather than an end-to-end approach; this has led to a greater ground infrastructure, but has not encouraged communications between space-based assets. This alone provides one of the key challenges that NASA must encounter. With the development of the new Crew Exploration Vehicle (CEV), NASA has the opportunity to provide an integration path for the new vehicles and provide standards for their development. Some of the newer capabilities these vehicles could include are routing, security, and Software Defined Radios (SDRs).

To meet these needs, the NASA/Glenn Research Center's (GRC) Network Emulation Laboratory (NEL) has been using both simulation and emulation to study and evaluate these architectures. These techniques provide options to NASA that directly impact architecture development. This paper identifies components of the infrastructure that play a pivotal role in the new NASA architecture, develops a scheme using simulation and emulation for testing these architectures and demonstrates how NASA can strengthen the new infrastructure by implementing these concepts.

### **ABOUT THE AUTHORS**

**Richard Slywczak** is a computer engineer with the Satellite Network and Architectures Branch at the NASA/Glenn Research Center (GRC) in Cleveland, Ohio. He is involved with the Network Emulation Laboratory (NEL) that provides simulations and emulations of NASA-based communication systems. NEL is currently looking at the implementation of Delay Tolerant Networking (DTN) on the communications infrastructure and the associated impacts. He has also been involved in the development of the GRC Channel Emulator (CE) that provides space-based characteristics for simulating communication data links. In addition, he is part of the Space Telecommunications Radio System (STRS) Architecture team, which is developing the next-generation Software Define Radio (SDR) common infrastructure that standardizes the development and implementation of NASA waveforms. He holds a B.S. in Aeronautical Engineering from the Pennsylvania State University and an MS, Computer Science, from Johns-Hopkins University.

## Development of Network-based Communications Architectures for Future NASA Missions

**Richard Slywczak**  
**NASA/Glenn Research Center (GRC)**  
**Cleveland, Ohio**  
**Richard.A.Slywczak@nasa.gov**

### INTRODUCTION

Within the next decade, NASA will encounter a number of unique challenges that encompass effective and continuous communications between mission nodes. Externally, the agency must develop and launch a new human-rated vehicle that will replace the aging Space Shuttle, scheduled for retirement in 2010. What is not obvious is the expectation that the Crew Exploration Vehicle (CEV) – also known as Orion – will be integrated into a seamless, flexible, and manageable communication infrastructure. The major benefit a routable infrastructure provides to NASA is that communications patterns will no longer be highly scheduled and pre-determined, but satellites will dynamically determine where to route the data.

Currently, NASA uses a monolithic system for satellite communications given limited satellite-to-satellite access links. The notable exception is the Tracking and Data Relay Satellite System (TDRSS); a bent-pipe architecture where data can only be transmitted on a highly scheduled basis. (SNUG, 2002) TDRSS doesn't perform any notable end-point determination for the data, but instead relays the data to the entity where the High-Gain Antenna is pointed. To create routable infrastructures, NASA must define potential candidates for future space communications architectures. By using component-based subsystems, each of the data links can be classified as a link defined by a set of properties. The benefit of a component-based design is each link can be characterized and defined by the types of data protocols that will be transmitted, associated antenna types, and data rates.

NASA will require routable networks on future science and infrastructure missions, but has limited operational experience in these communication architectures for space missions. (NASA-Constellation, 2007). For Low-Earth Orbit (LEO)

environments, flexible data communications will be less problematic, but as missions extend to the lunar regime and on to Mars, some of the architecture and protocol selections have the potential of breaking down. (Slywczak, 2004). NASA/GRC has been working on the Network Emulation Lab (NEL) to specifically examine these types of architecture in a hybrid simulation/emulation environment. (Slywczak, HQ Workshop, 2006) A hybrid environment is extremely helpful, since the orbital characteristics are simulated very quickly, but communications aspects - the focus of NEL - are performed real-time so that realistic data can be obtained. This hybrid combination of non-real time and real-time simulations and emulations allows candidate communication architectures to be thoroughly tested and examined without incurring the cost or overhead for aspects that are not directly germane to architecture being emulated.

This paper provides an overview of the work that NASA/GRC has invested in the NEL, as well as the characteristics for each of the components and how these can be modeled. Given the amount of data that can be produced by the emulation system, a customized visualization tool was developed to help analyze the data and determine Targets of Opportunities (TOOs). This allows data can be segmented and analyzed independently from NEL.

### PROBLEM STATEMENT

An important concept is the NEL runs emulations, which are referred to as scenarios that are end-to-end rather than a point in time. The scenarios will run from a specified start time to a specified end time that can last hours to days. Scenarios contain components and entities that represent aspects of the architecture being studied. It permits a "what-if" type of analysis answer the following example questions:

- How many relay lunar satellite would be required to have 24-hour contact with

astronauts on the lunar surface? 18-hour contact? 12-hour contact?

- If less than 24-hour contact is assumed, where are the points that communications will drop out? How much time is considered lost?
- Which ground stations provides the most/least amount of communications contact? Based on the currently installed equipment, what is the throughput that can be expected? How much data can be transmitted to the ground station?
- How efficient is a new routing algorithm? Can it route data successfully between nodes that are part of the architecture?

To answer each of these questions, the architecture, such as number and capability of the satellites, ground stations, data links, etc., must be well established for that particular scenario. Therefore, the types of links and entities must be well defined as input to the emulation run. Each run can then tweak or redefine these elements to optimize or derive different results.

The goal of this paper is two-fold. The first is to show how the input is derived in terms of a sample space-based architecture. The second part will provide a brief overview of the emulation system.

### HYBRID EMULATION

Given that NASA is changing the basic communications infrastructure, limited mechanisms for verifying and validating space-based networking technologies exists. Besides, on-orbit testing is not realistic, given cost and schedule, until these technologies have been proven on the ground to their maximum extent. The justification and goal of ground to on-orbit testing is a basic fact that every NASA technology must endure. NASA/GRC NEL is a unique capability that can take the components of satellite-based architectures and models these based on the characteristics of the components. For example, if an antenna is being modeled, it can emulate the data throughput characteristics with data and data rates and the appropriate protocols.

Figure 1 shows the progression of systems development using Simulation, Emulation, and Field Testing. In the figure, the model progression is from coarse grain to fine grain, as requirements and model development are improved through simulation, emulation, and testing. The model is cyclic where, once field-testing has been completed, refined requirements can be placed into the simulation

module and iterate through the each of the steps until a definitive list of refined requirements exist.

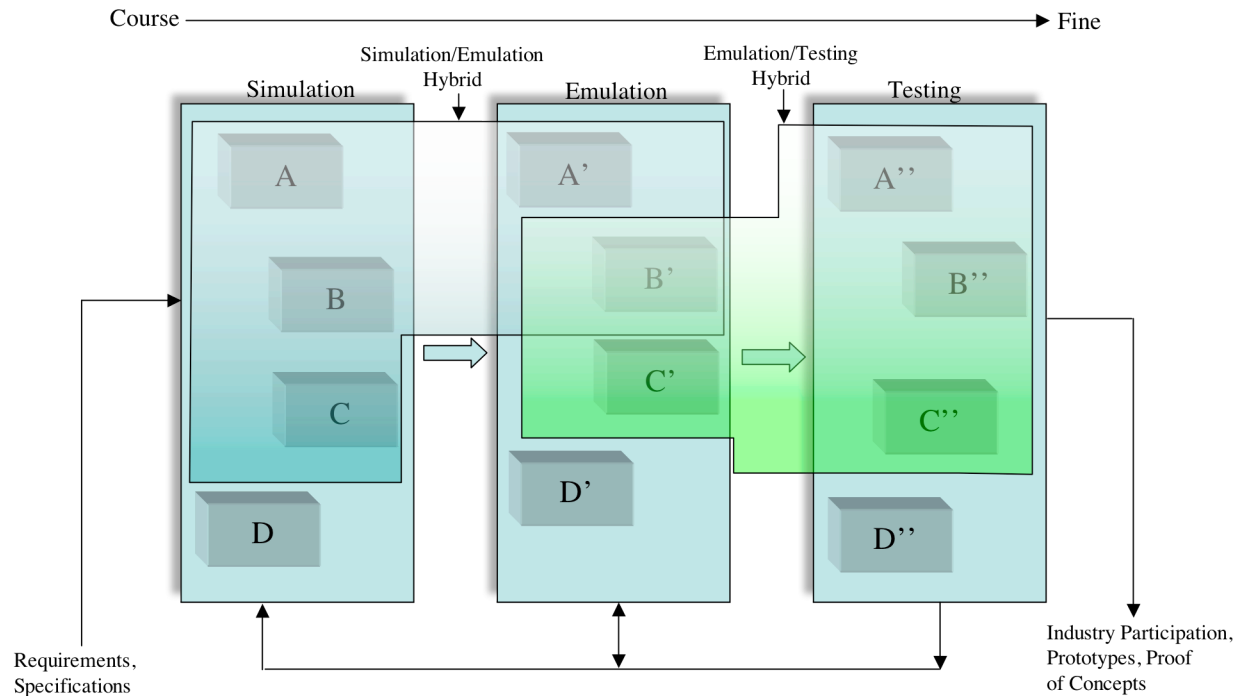
The NEL is focused on the hybrid simulation/emulation area of diagram where aspects of both simulation and emulation are introduced into the model. This is shown in the diagram under the arrow indicated by Simulation/Emulation Hybrid. Since the goal for NEL is to run in real time, aspects that are not related to communications, such as orbital analysis, assess time for entities, or delay time profiles, can run in simulation either before or during the emulation run, as long as the data is available when needed. Each of these analyses will feed into the emulation system as required. This provides a highly adaptable system and permits the end user to achieve results in a time efficient manner as possible.

### SPACE COMMUNICATIONS ARCHITECTURES

Figure 2 shows a representative and notional space-based architecture that contains the necessary elements for an end-to-end diagram. The goal for this figure is to show as many representative data links and elements as possible.

As mentioned previously, end-to-end scenarios can be demonstrated through this figure. Once a scenario is input into the emulation system, either the complete scenario or parts of the scenario can be used. The concept of running through that complete architecture from a source to a destination is described as an end-to-end scenario. For example, the CEV can be launch on a mission into lunar orbit. For this scenario, it is assumed that a lunar base exists on the Moon. The steps for this scenario are defined, as follows:

- The CEV would be launched from Kennedy Space Center (KSC).
- The CEV orbits the Earth two in LEO. The CEV would need to maintain contact with the ground Earth stations for voice communications, telemetry and other associated data. The emulation can show potential and actual communication contact. Potential contact indicates when a communication contact exists and actual is when data is flowing over the data link.
- The CEV would then make a Trans-Lunar Injection (TLI) on its path to the Moon. During this maneuver, the CEV would maintain contact with both the Earth ground stations and the lunar base on the Moon. This is an aspect where delays can affect the data communications, since, as the CEV



**Figure 1: Simulation, Emulation, and Testing**

moves farther from the Earth, the delay will increase and as the CEV approaches the Moon, the delays will decrease. The emulation system must account for these by processing the time delay profile.

- Once the CEV reaches the Moon, it will insert itself into lunar orbit and communicate with both the lunar and Earth ground stations.

Since the goal is to achieve autonomous operations, which will be required when NASA approaches Mars missions, this scenario reasonably determines how the CEV can transmit data to the Earth without using a prescheduled link. This requires that a router on-board the CEV will determine the next best neighbor to transmit data based on access time, neighbor power, neighbor preparedness, and other aspects. The neighbor can be part of a relay system or a science mission with routing capabilities.

A valid scenario, at a minimum, must contain two entities and a data link that connects them. Therefore, a scenario could be a satellite in LEO that communicates with a ground stations or a satellite in lunar orbit that transmits data to a satellite in LEO.

### Object-Oriented Development Paradigm

Each of the components of the space-based architecture is developed using an object-oriented

method where they are defined as an entity with a set of functions and characteristics. For example, a link can be defined with characteristics such as delay and bit error rate profile and functions, such as activate or deactivate. An antenna can be defined with characteristics, such as data rate, rate of change for antenna movement, and current direction. Some of the functions that might be placed on an antenna would include moving the antenna, changing the antenna type, and providing transmission capabilities.

The current set of components is defined in the following two sections, but users are not limited by the current components and can extend those capabilities as they develop scenarios. For example, the user might want to define a new antenna rather than using the current definition by inheriting from a basic antenna and defining new functionality.

The following two sections will discuss the types of entities and link characteristics.

### Entities

The communications architecture will contain a number of entities that are active at defined points in time in a communications scenario. These entities will be the main data transmitters and receivers over predefined links. An overview of the entities that are currently part of the emulation are described as follows:

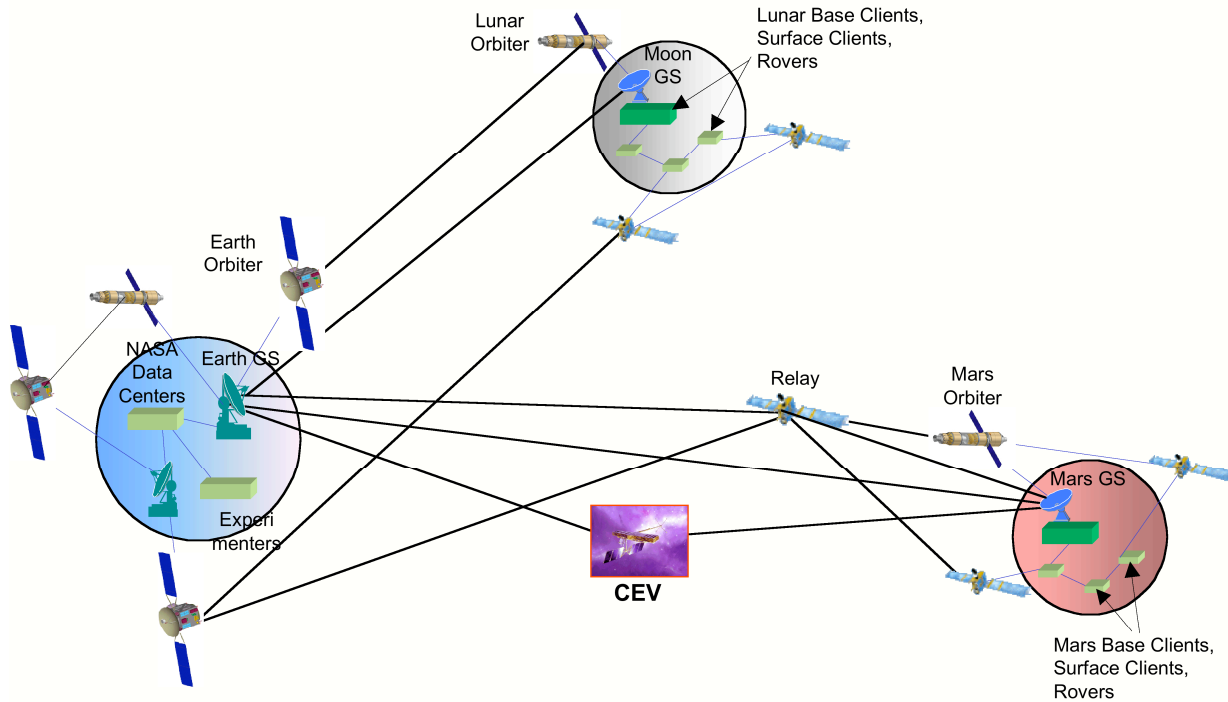


Figure 2: Sample Scenario with Entities and Data Links

- Satellite:** A satellite is an entity that orbits a central body. The satellite can represent a human-rated or non-human rated vehicle in the current emulation system. In future revisions, these might be unique as requirements concerning the two are refined. The satellite will be characterized by its orbit (LEO, GEO, lunar, etc) and, if possible, can move between central bodies. Currently, all orbital characteristics are obtained from a Satellite Toolkit (STK) simulation. As discussed in a later section a satellite is contained on a single computer system and must contain the required protocols.
- Antenna:** An antenna will be a communications device between two entities that includes both satellites and ground stations; therefore, an antenna cannot exist by itself, but must be connected to another entity. Antennas include both unidirectional which can communicate without being pointed and directional antennas that require pointing. Algorithms that govern the antenna, such as movement, pointing, modulation, etc., can be modified within the NEL.
- Instruments:** Instruments are data collecting or transmitting devices, such as scanning instruments, cameras, or radios that are connected to an entity. An instrument cannot exist by itself. Currently, an instrument is not required for a satellite to be instantiated, since all vehicles or ground stations are capable of transmitting data. In a future revision, communication devices will have to be explicitly added.
- Base Station:** A base station is an entity on a central body where experiments or instruments are placed. To communicate, a base station must be connected to – or transmit data through - a ground station that provides the antenna characteristics. The base stations will form the future SpaceHabs that will be part of the lunar ground communications.
- Ground Station:** A ground station is a ground entity on a central body that can host an antenna. Ground stations can exist by themselves or they can be connected to a base station. They receive data from an orbiting entity that is destined for the ground.
- Central Body:** A central body is the planet about which a satellite orbits or ground stations are based. Currently defined central bodies in NEL are the Earth and Moon, since that is the current NASA focus. Eventually, Mars will be added to the emulation system.

## Data Link Types

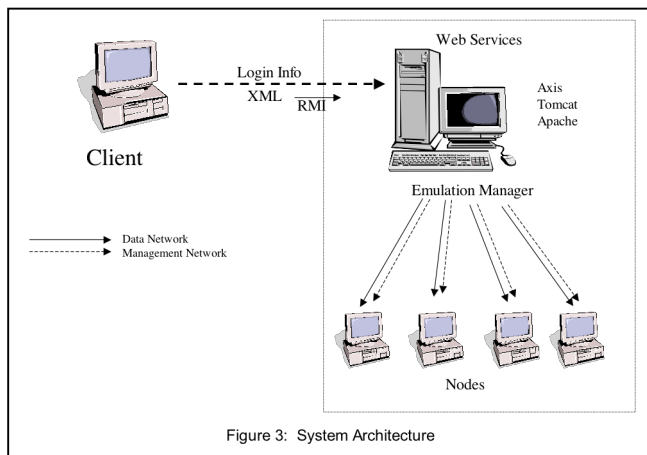
The communications infrastructure is composed of a number of smaller segmented data links that will be characterized during data transfer, as shown in Figure 2. The data links connect multiple entities in a scenario.

For example, a satellite can be connected to another satellite through an inter-satellite data link. Characterizing these links aid in communications and protocol development, since specifying links for a specific mission and interfacing with the remaining infrastructure enhances the goal of becoming a cookbook approach to design. If a specified link is required, then the data rates, specified protocols, security – given the specified data type - and other infrastructure considerations can be specified. Importantly, while there is never a “one-size fits all” solution, a majority of the links can be specified using existing characteristics, but the objective of enhancing the data links to include new aspects will be the overall goal.

Current NEL links can be summarized as follows (Bhasin, 2001):

- **Backbone Links:** Backbone links represent long-haul, long-duration links that will sustain the maximum impact of the space environment, in terms of delays, bit error rates, duplication, losses, etc. Delays on backbone links are a minimum of 1.5 seconds and can easily exceed minutes when considering planets like Mars and beyond. Most likely, these links will connect interplanetary relays, but could also be a Deep Space antenna sending emergency commands to a satellite in “safe-hold” orbit around another planet. Examples of these links represent ones that would emanate from an Earth-based backbone transmitting to a lunar or a Martian-based backbone, or, likewise, from a Martian-relay to an Earth-based relay. Since all interplanetary traffic would need to traverse these links, they will be highly optimized in high delay environments and contains such protocols as Delay Tolerant Networking (DTN) that also provides the concept of store-and-forward, when data links are disconnected. (Warthman, 2003). Given the delay time, most of the data that traverses these links would be in an unreliable protocol and, those requiring a reliable protocol would implement a post-traversal validation scheme.
- **Inter-Satellite Links:** Inter-Satellite Links promote communications between satellites for transferring data within a planetary environment. These links are characterized by short-delay, short-duration links, which could range in delay up to approximately 0.5 seconds. These links would not be as concerned about delays, bit error rates, etc., since traditional reliable protocols are applicable, but are still concerned about disconnected networks. On an inter-satellite link, the source entity must determine the most appropriate mechanism for transferring the data and, if one entity is not present, it has the ability to choose another. Similarly, multiple entities could be present and the source entity might need to decide, based on transmit power, length of access (i.e., visibility) or a number of other characteristics, which links to use. The receiver might not be part of the mission, but would be willing to accept data to route to the next hop in the path to the destination. The receiver could be part of a relay constellation or a single satellite that has routing capabilities. If no receiver were present, then a store and forward mechanism, such as DTN would need to be implemented. Transmissions on these links could either be reliable or unreliable.
- **Access Links:** Access links are qualified as links between the outer edges of a backbone network and spacecraft, mission vehicles and other entities around the surface of a planet. These links are specialized inter-satellite links where the communications is between the backbone node and some intermediate space-based node. Delay times for access links are nominally less than 0.5 seconds. Access links would handle traffic that is traveling through the interplanetary network and is re-entering the network within a planet. For example, if data were traversing the network from Earth to Mars, it would leave the Earth-based backbone network and enter the Mars-based network through a Martian backbone network. The traversal from the backbone network to the Martian WAN is through an access link. This case is distinct from an inter-satellite link because there may be particular cases to handle, such as in an Earth-based scenario, from a geosynchronous to a LEO satellite in terms of packet corruption. Transmissions on these links could be reliable or unreliable.

- **Proximity Links:** Proximity links are direct to surface (DTS) or those that originate at a space-based entity and are destined for the surface. Given that, by definition, proximity links are short duration and low delay, most protocols would be valid for the implementation of these links. In the Earth-based networks, a proximity link would be from a LEO satellite to a ground station. Proximity Links have delay times less than 0.5 seconds. Proximity links are the second portion of a specialized inter-satellite link where the access and proximity link create a specialized inter-satellite link.
- **Planetary Links:** The Planetary Links are ground-based networks on a planetary surface, which consist of Wide Area Networks (WANs), Local Area Networks (LANs), Personal Area Networks (PANs), etc. While Earth-based planetary networks are well known, there are still issues and challenges associated with planetary networks on the lunar and other planetary surfaces due to atmospheric effects, etc. In addition, these links are required to create end-to-end scenarios.

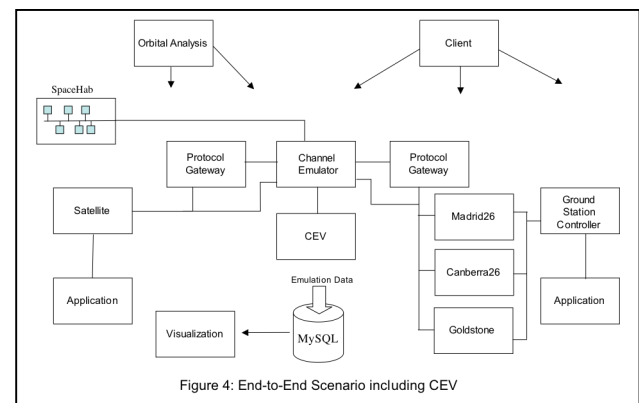


## EMULATION SYSTEM OVERVIEW

The NEL provides emulation services through a series of networked computers; the origins of NEL trace back to the Protocol Research Evaluation Environment (PREE) which was originally developed to test and evaluate different protocols, not only providing a validation and verification environment, but to emulate a network environment of multiple nodes. Since the system is scalable, the number of processing nodes is not system dependent, but scalable to include not only the number of emulations

currently running on the cluster, but also the complexity of the scenarios. NEL is capable of dynamically dispatching the number of nodes to execute a scenario, providing they are available. Given that the NEL's mantra is open-standards, open-interfaces, each node is addressable through a public<sup>1</sup> set of interfaces offered through web services. Figure 3 shows the schematic of the system architecture and Figure 4 shows a specific instance of an emulation run.

This brief overview of NEL will be divided into a number of distinct sections. The first will discuss the system architecture of NEL and how the cluster is configured. The next will provide information on the input to the system, which covers the definition of scenarios and how the information is conveyed to each processing node. The duties and configuration of the processing nodes are covered and then the data collection from each of the nodes into portable formats, which becomes input for the visualization program.



## System Architecture

The guiding goal for the development of the NEL was to create a portable and modular system so that it can be distributed to the users home environment rather than being centrally hosted. This not only reduces the resources required at the central site (i.e., the NEL laboratory), but also subscribes to the philosophy that a powerful and flexible system should be distributed rather than centralized. As described in the following sections, the input to the system is through a GUI scenario builder and the output analysis tool is also through a visualization tool. In addition, the Channel Emulator (CE) can be distributed for use in other testbed environments and

<sup>1</sup> In this case, public implies that these services are only publicly available to the emulation engine on the cluster and not public to the terrestrial Internet.

has been installed and used at other NASA Centers and Universities. Currently, the only part of the emulation system required at the NEL is the emulation engine.

Internally, an emulation manager controls the cluster, which is responsible for job distribution among the nodes in the cluster. It takes as input, an XML file, which contains the elements required to completely define the elements in the scenario. The XML file is an output of the scenario builder GUI program.

The emulation manager is also responsible for coordinating the simulation programs that provides input to the emulation system. Currently, the simulation program is STK, which provides orbital characteristics that the emulation manager must distribute to the nodes. Any other simulation, such as OpNet (OpNet, 2007) or QualNet, (QualNet, 2007) could also serve as input to the emulation manager. Once the emulation manager has divided the entities between the nodes, it will subdivide the XML file into the appropriate sub-modules, based on the scenario. For instance, if a processing node is allocated as a specific ground station, for example White Sands, then the XML configuration file that describes White Sands will go to that node.

In addition, the emulation manager is also responsible for configuring the data links between the nodes. It provides configuration information to the Channel Emulator about the delays, bit errors rates and other required information.

Once the emulation has started, it will be responsible for keeping the nodes synchronized. It must ensure that the time of each of the entities is the same time across the entire cluster.

In addition, the emulation manager is responsible for health and safety monitoring of the cluster and preemptively reports any issues with the system, such as disk space problems, memory issues, to the controller.

### **Processing Nodes**

The nodes are individual computers that emulate portions of the scenario and are segmented with respect to functionality, such as a lunar-orbiting satellite or a ground Earth Station. The node is required to parse the XML file that was supplied by the emulation manager and configures itself for the processing run. Each node contains two interfaces: management network interface and the data network interface. Having two interfaces keeps the

management data separate from the actual data being sent over the emulated satellite network. This allows for more accurate modeling of the data links by not corrupting the data stream.

Each processing node in the cluster can be allocated a specified set of tasks that defines a single entity for the scenario. For example, if a node is configured to be a satellite, then it must perform the processing that is representative of a satellite. These include a standard set of services, such as geolocation processing and antenna control, and instance specific services, such as data routing. Each node will be timed and synchronized by the emulation manager.

### **Space-Link Characteristics**

Space link characteristics are injected by a GRC-developed Channel Emulator (CE) that injects delays, bit error rates, and packet mangling (loss, corruption, duplication, etc.) to all data that traverses between the ingoing and the outgoing interfaces. (Slywczak, 2006). The CE functions as a Level 2 link layer device so that it has the option of delaying other protocols besides the Internet Protocol (IP), i.e., the CE is IP independent. It runs on a separate computer with each of the nodes connected to an interface on the CE. Each physical Ethernet card is viewed as a separate antenna on the vehicle that the node is emulating where the data rates and antenna characteristics can be set within the CE.

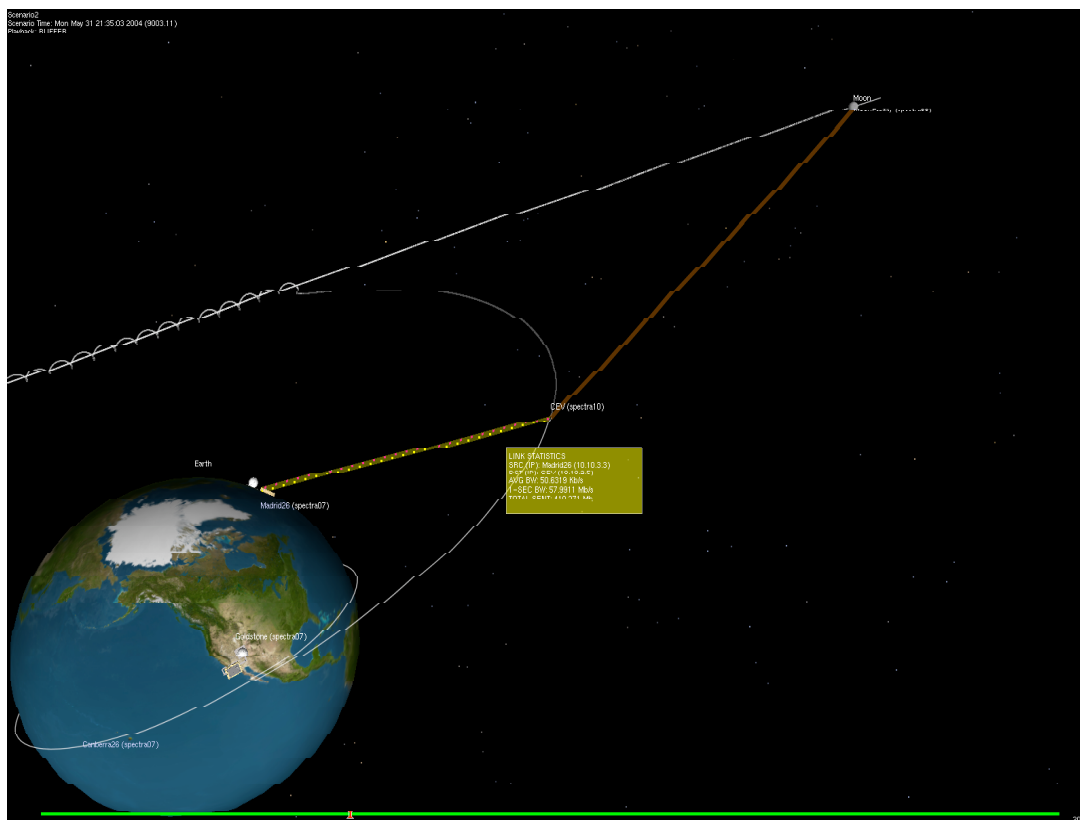
The CE is based on the open-source package called netem that provides the core for the delays and bit errors, and the GRC-developed software provides for ease of configuration and the option for using Virtual LANs (VLANs); VLANs divide a single physical interface into multiple virtual interfaces. The CE has the option to take a time-delay and bit error rate profile from STK and delay all data based on this profile during an emulation run.

All data collection, as discussed in the next section, will happen in the CE.

### **Data Collection**

Importantly, the emulation system runs at real time, so, during the run, it will record a significant amount of information. Currently, data recording occurs at the CE, so that every data that is transmitted between the entities (Earth to lunar, Earth to Mars, LEO to a Ground Station) is captured and, from this data, the visualization program is able to determine when connections are established, who made the connections, and how much data are transmitted.





### Figure 5: Scenario through the Visualization Program

Example information collected includes the following: Time, IP Header Information (such as Source Address, Destination Address, Protocol, Checksum, and Priority Bits), TCP Header Information (such as Source Port Number, Destination Port Number, Sequence Numbers, and Priority Bits), UDP Header Information (such as Source Port Number, Destination Port Number, and Packet Length), and ICMP Information (such as Type and Code).

This information can be written to a database file, currently MySQL, a pcap file, or a text file. Each format provides a portable format to third-party applications and the visualization program can ingest data from either a MySQL database file or a text file.

## Visualization

The visualization program is responsible for taking the output from the emulation system and displaying it to the user. The visualizer plays an important role in simulations and emulations providing a tool for data mining. For space exploration, it is even more vital because of the complex geospatial nature of the space environment and networking statistics.

The NEL environment provides user with an advanced 3D visualization capability. It is a multi-platform, stand-alone application that can run in real-time with an on-going emulation or in playback mode. With the playback capability, remote users can see the emulation at any time without being present in the testbed. The visualizer is written in C++ on top of OpenGL (graphics) and glut (windowing) libraries.

The basic functions of this visualizer are to 1) display all space entities in a 3D view and provide a means for navigation, view saving and restoration, 2) show orbits and trajectories of satellites, 3) show possible connection between entities and data communications, 4) show network statistics of an active link including the type of data, direction, protocol used, throughput, bandwidth, delay, and bit error rate and 4) show history of access.

In playback mode, the visualizer can playback a previously run scenario and provide users with basic Fast-Forward/Rewind, Pause/Resume functions. A remote user can connect to the emulation at any time during the run and visualize the emulation data real-

time from the start of the scenario to the present processing point.

### **FUTURE WORK**

This discussion of NEL is a snapshot of the current implementation rather than a final system. While it currently provides an operational system with its current capabilities, there is still a significant amount of work left to accomplish. Some of the work is awaiting decisions by NASA, such as design directions for the CEV. Future work for NEL is summarized as follows:

- Fine-tune and extend user applications. Both the scenario development and the visualization tool are considered prototype software. Conceptually, they are distributed to the end users for test and evaluation, and NEL has received comments on the tools. But, to be extremely effective, they must be configured and maintained independently of the configurations developed within NEL.
- Hardware-in-the-Loop (HWiL) Integration. One of the limitations of the current NEL is that all scenarios are required to be software-based. Therefore, if a new radio, router, or other hardware device is developed, the device cannot be integrated into the current testbed. However, a node could be configured to look like the device and the software algorithms independent of the hardware could be integrated into the testbed. In a future version of the testbed, true HWiL will be considered and implemented.
- Satellite Implementation. In the entity discussion, the satellite is considered to be amorphous and no differentiation is made between human-rated and non human-rated vehicles. As more definitions and decisions are made regarding the CEV, this differentiation will need to be apparent in the NEL. The degree and implementation of this change will be based on the development of future human-rated vehicles within NASA.
- Extend the component-based development. The NEL is considered to be a communications laboratory and not an operational facility. The goal is to adopt and extend research projects that can be implemented by other organizations. The end goal is to implement a distributed testbed environment where interested parties could download parts of the testbed to run at their facility. The best component-based

example would be the CE where it has been used outside of the emulation environment. The goal is to componentize the complete emulation environment so that it can be operationally implemented by organizations external to the laboratory.

### **CONCLUSION**

One of the realistic conclusions about NASA future development is the requirement for extreme modeling with simulation and emulation support. The issue is that as budgets and schedules become tighter, simulation and emulation provide a mechanism for testing and verifying advanced concepts before the development of flight missions or even pressing space-qualified parts.

With the radical changes to the communications paradigm being discussed at NASA, highly scheduled, dedicated, circuit-switched based links will give way to a flexible and modular routable infrastructure. But to implement such a system, NASA must redefine the way it develops and operates missions. In the end, missions should be less expensive to develop and operate given more upfront planning and also access to infrastructure that will be available in the future.

NASA/GRC has proposed using the Network Emulation Lab (NEL) to perform advanced network modeling. As discussed in this paper, the end-to-end architectures must first be developed. The proposed method is to divide these architectures into a set of well-understood components that can be presented to the user. Using the components of entities and space data links, scenarios can be developed that serve as input to the emulation system. However, each of these components can be highly modified, so that the emulation run can serve as a "what-if" analysis tool for the user.

The second part of the paper provides a brief overview of the NEL. The emulation manager commands the emulation systems by providing configuration and synchronization services to each of the nodes in the computation cluster. It also monitors the cluster for any issues or problems and will alert the operator.

The cluster is divided into a set of responsibilities. Beside the emulation manager, each processing node will represent an entity which can be a satellite, ground station, etc. All processing nodes are connected to the GRC Channel Emulator (CE) to model the space-based data links in terms of delays,

bit error rates, loss, etc. All data is collected by the CE and placed into a data format that can be read and displayed by the customized visualization program that permits the user to look for events of interest.

The NEL provides a flexible, cohesive system for advanced modeling of future communication systems. As NASA reconsiders its model for space-communications, they will need to do extensive simulation and emulation to ensure that correct solutions are chosen and implemented.

## REFERENCES

- Bhasin, K, Hayden, J., Space Internet Architectures and Technologies for NASA Enterprises, IEEE Aerospace Conference, 2001
- Holtz, A., Bhasin, K., et al., Evaluating Space- Based Protocols through the Protocol Research Evaluation Environment, AIAA Modeling and Simulation Conference, 2005
- NASA – Constellation Program: NASA’s New Spaceships from [http://www.nasa.gov/mission\\_pages/constellation/main/index.html](http://www.nasa.gov/mission_pages/constellation/main/index.html), accessed June 2007
- OpNet Technologies from <http://www.opnet.com>, accessed June 2007
- Slywczak, Richard A., Low-Earth-Orbit Satellite Internet Protocol Communications Concept and Design, NASA Technical Memorandum, 2004
- Slywczak, Richard A, Kollar, Thaddeus, Modeling Space-Based Communications using a Dynamic Channel Emulator, AIAA Modeling and Simulation Conference, 2006
- Slywczak, Richard A., Space Network Simulation/Emulation Testbed – A Technology Overview, NASA HQ Workshop, 2006
- Space Network Users Guide (SNUG), NASA/Goddard Space Flight Center, 2002
- QualNet from <http://www.scalable-networks.com>, accessed June 2007
- Warthman, Forrest, Delay Tolerant Networking (DTNs): A Tutorial 1.1, March 2003